UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/580,557	05/26/2006	Benjamin Liu	42P22191	5968

45209 7590 01/20/2011

MISSION/BSTZ
BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
1279 OAKMEAD PARKWAY
SUNNYVALE, CA 94085-4040

EXAMINER	
WILLIS, JONATHAN U	

ART UNIT	PAPER NUMBER
2445	

MAIL DATE	DELIVERY MODE
01/20/2011	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/580,557	LIU, BENJAMIN	
	Examiner	Art Unit	
	JONATHAN WILLIS	2445	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 November 2010*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,2,4-7,9-11,13,15-18,21 and 23-39* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,4-7,9-11,13,15-18,21 and 23-39* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *26 May 2006* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☒ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This Office Action is responsive to the Arguments/Remarks filed on 11/12/2010. No claims have been amended. Claims 1, 6-7, 9-11, 18, 21, 23-26, and 32 have been amended. Claims 3, 8, 12, 14, 19, 20, and 22 are cancelled. Claims 1-2, 4-7, 9-11, 13, 15-18, 21, and 23-39 are pending examination.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claims 6-10, 18-19, 21 and 23-25 are rejected under 35 U.S.C. 101 because the claimed invention is directed to nonstatutory subject matter.

The claims are drawn to a "tangible computer readable medium." The specification is silent regarding the meaning of this term. Thus, applying the broadest reasonable interpretation in light of the specification and taking into account the meaning of the words in their ordinary usage as they would be understood by one of ordinary skill in the art (MPEP §2111), the claim as a whole covers both transitory and non-transitory media. A transitory medium does not fall into any of the 4 categories of invention (process, machine, manufacture, or composition of matter).

To overcome this rejection, Examiner suggests changing “tangible computer readable medium" to "non transitory tangible machine readable medium, thus excluding

that portion of the scope covering transitory signals. The scope of the disclosure given the state-of-the-art covers both transitory and non-transitory media, and this amendment would limit the claim to an eligible (non-transitory) embodiment.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claim 1-3, 6-7, 11, 18 and 32 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bantz (US2006/0107269) in view of US2004/0167996 A1 to Takamura et al. (hereinafter referred to as Takamura).

6. In regard to claim 1, **Bantz** teaches a method for a client platform coupled to a server platform via a network (*see client coupled to server via network,* **in Fig. 3 [101] [104]**) comprising:

determining (*e.g. "recognized,"* **in [0006] Line 3**) that an input/output operation (*e.g. "plugged in,"* **in [0006] Lines 2-3**) related to an input/output device (*e.g. "devices local to the user to be "plugged in", recognized,"* **in [0006] Lines 2-3**) happens during execution of an application on a virtual machine (*e.g. "devices local to the user to be*

"plugged in", recognized, and made available to the user while executing on the remote virtual machine," **in [0006] Lines 2-4**),

requesting the server platform via the network to handle the input/output operation (*e.g. "The virtual device hub senses that a device has been plugged into the hub in step 202, gathers the information about the device...The device information is used...to find out if support for that particular device exists on the server 101,"* **in [0027] – [0028]**),

wherein the request (e.g. "*find out if support for that particular device exists on the server,"* **in [0028] Lines 3-4**) comprises a device module identifier to identify a device module (e.*g. "gathers the information about the device such as the device model number and type, and sends that information to the virtual machine instance in server,"* **in [0027] Lines 5-8**) from a plurality of device modules (s*ee inherent searching through multiple device drivers*, *e.g. "the device driver to be located*," **in [0006] Line 7**) in the server platform to handle the input/output operation related to the input/output operation (*e.g. "find out if support for that particular device exists on the server…If not, the virtual machine instance in the server initiates the installation of a physical device driver in the server,"* **in [0028] Lines 3-6**), wherein the device module is a virtual device corresponding to the input/output device (*e.g. "The device information is used to allow the virtual machine to choose what type of device 103 has been inserted into the virtual device hub...the virtual machine instance in the server 101 initiates the installation of a physical device driver in the server 101 and a virtual device driver in the virtual machine instance executing on the server 101,"* **in [0027] Lines 5-8**), but

Bantz does not explicitly teach that

the virtual machine is run on the client platform; and

requesting the server platform via the network to handle the input/output operation related to the input/output device is through a client network interface of the client as claimed.

However, **Takamura** teaches

the virtual machine (*see guest operating system ran in client,* **in Fig. 2 [122]**, *e.g. "The startup processing 320 is called when the client computer 101 is started and it activates the hypervisor and the OS,"* **in [0045] Lines 4-6**) is run on the client platform (*see "Client Computer,"* **in Fig. 1 [101]**); and

requesting the server platform (*see "Server Computer,"* **in Fig. 1 [102]**) via the network (*see "Network,"* **in Fig. 1 [103]**) to handle an input/output operation related to an input/output device through a client network interface (*see "Network Interface Adaptor,"* **in Fig. 1 [902]**) of the client (e.*g. "hypervisor of the client computer…for detecting an access to an I/O device of the server computer…and…transmitting a command to the I/O device of the server computer….A hypervisor of the server computer…which receives the command to the I/O device from the network, and issues the command to the I/O device,"* **in [0010] Lines 4-14**).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the claimed invention to add the feature of running a Virtual Machine Monitor (VMM)/Hypervisor in a client computer to detect I/O requests that need to be handled by the Hypervisor of a host systems, as disclosed in **Takamura**, into the teachings of

Bantz, since both reference are directed toward I/O operations of virtual devices, hence would be considered to be analogous based on their related fields of endeavor.

One would be motivated to do so as it is well known and old that virtual machines are run locally on client machines as well as remotely on host machines, depending on the application requirements, and **Bantz** also discloses that the virtualization of devices is also applicable to physical machines as well, wherein a virtualization layer/Hypervisor is run on top of the Operating System and interposed between the server and the peripheral devices in order to detect requests from a physical device such as a fat client (*e.g. "The invention is applicable to physical machines as well as virtual machines as shown in FIG. 3. When the invention is applied to a physical machine, a device virtualization layer 301 is interposed in the operating system between the device 103 and the server 101,"* **from Bantz in [0033] and** *e.g. "The layer, in turn, implements communication with real devices through local device drivers or through device drivers that communicate with the device virtualization layer of some remote platform,"* **from Bantz in [0012] Lines 6-9**), and one of ordinary skill in the art would recognize that clients and servers may run different operating systems, and **Takamura** discloses the need for compatibility between Client/Server I/O operations where different Operating Systems are being utilized on their respective platforms (*e.g. "there is a problem that in a computer system comprising a server computer and a client computer, connected via a network, when an OS of the server computer and an OS of the client computer are different from each other, an I/O device connected to the server computer cannot be used from the client computer,"* **from Takamura in [0008]**), and **Takamura** would

enhance **Bantz's** physical machine embodiment by allowing client's to locally run any operating system and use I/O devices that are not installed on a Virtual Machine being executed by a client (*e.g. "to allow the client computer to use an I/O device connected to the server computer, without changing the operating systems on any of the server computer and the client computer, even when those operating systems are different from each other,"* **from Takamura in [0009]**), thereby increasing the compatibility of the system.

7. In regard to claim 2, **Bantz-Takamura** teaches the method of claim 1, wherein the request (e.g. "*find out if support for that particular device exists on the server,"* **from Bantz in [0028] Lines 3-4**), comprises a server platform identifier to identify the server platform (*see inherent identification of server platform in connection of client,* **from Bantz in Fig. 3 [101] [104]**).

8. Claims 6-7 are corresponding machine readable storage medium claims (*see "HDD,"* **in Fig. 1 [903] [913],** *e.g. "In the HDD 903, there are stored an application program 121, an operating system 122, a hypervisor 123, and a boot loader 124,"* **from Takamura in [0028]**) of method claims 1-2; therefore, they are rejected under the same rational.

9. Claim 18 is a corresponding machine readable storage medium claim of method claim 11; therefore, it is rejected under the same rational.

10. Claim 32 recite limitations substantially the same as the limitations of claims 1 and 11; therefore, they are rejected under the same rational.

11. Claims 4-5, 9-10, 15-17, 23-31 and 35-39 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bantz-Takamura in view of US 4,860,190 to Kaneda et al. (hereinafter referred to as Kaneda).

12. In regard to claim 4, **Bantz-Takamura** teaches the method of claim 1, further comprising:

receiving a feedback for the input/output operation (*e.g. "the device to be detected locally,"* **from Bantz in [0006] Lines 6-8**) from the server platform through the network (*see installation as feedback, e.g.* "*downloaded, and installed to the virtual machine,"* **from Bantz in [0006] Lines 6-8**), but

Bantz-Takamura does not teach the feedback comprising a virtual machine identifier to identify the virtual machine in the client platform that is executing the input/output operation; and sending the feedback to the virtual machine identified by the virtual machine identifier as claimed.

However, **Kaneda** teaches the feedback comprising a virtual machine identifier (*e.g. "receives the identification number,"* **in Col. 6, Line 1**) to identify the virtual machine in the client (e.g. "*computer system,"* **in Col. 1, Lines 63-65**) platform that is executing the input/output operation (e.g. "*computer system for controlling virtual*

machines, each machine given a different identification number," **in Col. 1, Lines 63-65**); and

sending the feedback to the virtual machine identified by the virtual machine identifier (*e.g. "to control the virtual machines and to decide which virtual machine will receive the control right of the CPU. The VM monitor assigns the identification numbers for the virtual machines,"* **in Col. 5, Lines 55-59**).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the claimed invention to add the feature of multiple virtual machines with different identification numbers as disclosed in **Kaneda** into the teachings of **Bantz-Takamura** since all of the references are directed to virtual machine operating system environments, hence, would be considered to be analogous based on their related fields of endeavor.

One would be motivated to do so in order to specify which virtual machine running on the client is to receive feedback, as it should be obvious to one of ordinary skill in the art to recognize that some sort of identification is necessary when transferring data in a network to a particular endpoint that has a plurality of equivalent environments for that endpoint, as **Takamura** also discloses the use of multiple guest Operating Systems in as single computer platform (e.g. "*In an actual computer system, however, there are many cases that such an OS-based I/O device virtualization function is unusable. This is because the I/O device virtualization function is available only between identical operation systems, in many occasions, and further, a plurality of types*

of OS are mixed in one computer system in general," **from Takamura in [0007]**).

13. In regard to claim 5, **Bantz-Takamura** teaches the method of claim 1, and receiving instructions via the network (*e.g. "Mouse movements are tracked at the user's local machine and sent to the remote virtual machine via the network,"* **from Bantz in [0010] Lines 5-7**), and a device module of the server platform (e.g. "*The device information is used to…find out if support for that particular device exists on the server,"* **from Bantz in [0027] Line 7 – [0028] Line 4**), but

Bantz-Takamura does not teach the method further comprising:

receiving an interrupt instruction issued by a device module, the interrupt instruction comprising a virtual machine identifier to identify a virtual machine to perform the interrupt instruction; and

injecting the interrupt instruction into the virtual machine identified by the virtual machine identifier

However, **Kaneda** teaches the method further comprising:

receiving an interrupt instruction (*e.g. "if an interrupt request is in that port, an I/O interrupt for the VM monitor of the real machine will be generated,"* **in Col. 4, Lines 20-22**) issued by a device module (*e.g. "I/O interruption queue,"* **in Col. 4, Line 19**), the interrupt instruction comprising a virtual machine identifier (*e.g. "identification number,"* **in Col. 6, Line 1**) to identify a virtual machine to perform the interrupt instruction (*e.g. "By this handling routine…it is determined which virtual machine has issued the I/O instruction which caused the I/O interrupt,"* **in Col. 6, Lines 40-43**); and

Injecting the interrupt instruction (*e.g. "By this handling routine," in Col. 6, Line 40*) into the virtual machine identified by the virtual machine identifier (*e.g. "By this handling routine…it is determined which virtual machine has issued the I/O instruction which caused the I/O interrupt,"* **in Col. 6, Lines 40-43**).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the claimed invention to add the feature of using VM identifiers with interrupts as disclosed in **Kaneda** into the teachings of **Bantz-Takamura** since all of the references are directed to virtual machine operating system environments, hence, would be considered to be analogous based on their related fields of endeavor.

One would be motivated to do so because it is well known in the art that I/O interrupts with identifiers are used to detect I/O operations in physical and virtual environments, and **Kaneda** discloses the need for a VMM to monitor and handle interrupts between multiple virtual operating systems (*e.g. "System control of the virtual machine is carried out by an operating system of the virtual machine. However because such control affects another virtual machines, the efficiency of the entire real machine system is decreased by interposition of the VM monitor, etc,"* **from Kaneda in Col. 1, Lines 25-29**), and **Bantz-Takamura** is enhanced by providing differentiation for requests between one of multiple Operating Systems (*e.g. "Virtualization of an I/O device is available only between the identical operating systems, because a method for I/O device virtualization varies with each type of the OS,"* **from Takamura in [0007]**).

.

14. Claims 9-10 are corresponding machine readable storage medium claims (*see "HDD,"* **in Fig. 1 [903] [913],** *e.g. "In the HDD 903, there are stored an application program 121, an operating system 122, a hypervisor 123, and a boot loader 124,"* **from Takamura in [0028]**) of method claims 4-5; therefore, they are rejected under the same rational.

15. In regard to claim 11, **Bantz** teaches a method for a server platform coupled to a client platform via a network (*see client coupled to server via network,* **in Fig. 3 [101] [104]**),

receiving, from the client platform via the network, a request for an input/output operation related to an input/output device (*see sending and receiving via network,* **in Fig. 1**, *e.g. "sends that information to the virtual machine instance in server…The device information is used to…find out if support for that particular device exists on the server,"* **in [0027] Line 7 – [0028] Line 4**) by a server network interface of the server platform (*see output sent to client device through inherent server interface, e.g. "The output is then routed to the actual printer 103 through the network connection and the virtual device hub 102,"* **in [0029] Lines 9-10**),

wherein the request (e.g. "*find out if support for that particular device exists on the server,"* **in [0028] Lines 3-4**) comprises a device module identifier to identify a device module (e.*g. "gathers the information about the device such as the device model number and type, and sends that information to the virtual machine instance in server,"* **in [0027] Lines 5-8**) from a plurality of device modules (s*ee inherent searching through*

multiple device drivers, *e.g. "the device driver to be located,"* **in [0006] Line 7**) in the server platform to handle the input/output operation related to the input/output operation (*e.g. "find out if support for that particular device exists on the server…If not, the virtual machine instance in the server initiates the installation of a physical device driver in the server,"* **in [0028] Lines 3-6**), wherein the device module is a virtual device corresponding to the input/output device (*e.g. "The device information is used to allow the virtual machine to choose what type of device 103 has been inserted into the virtual device hub...the virtual machine instance in the server 101 initiates the installation of a physical device driver in the server 101 and a virtual device driver in the virtual machine instance executing on the server 101,"* **in [0027] Lines 5-8**),

identifying a device module (e.g. "*downloaded, and installed to the virtual machine,"* **in [0006] Lines 6-8**) from a plurality of devices modules in the server platform to handle the request (e.g. "*find out if support for that particular device exists on the server,"* **in [0027] Line 7 – [0028] Line 4**), the identified device module (e.g. "*downloaded, and installed to the virtual machine,"* **in [0006] Lines 6-8**) corresponding to the input/output device related to the input/output operation (*e.g. "the device to be detected locally, the device driver to be located, downloaded, and installed to the virtual machine,"* **in [0006] Lines 6-8**);

obtaining a result (*e.g. "recognized,"* **in [0006] Line 3**) for the input/output operation (*e.g. "the device to be detected locally,"* **in [0006] Lines 6-8**) from the identified device module (e.g. "*downloaded, and installed to the virtual machine,"* **in [0006] Lines 6-8**);

constructing a feedback with the result (*see installation as feedback, e.g.* "*downloaded, and installed to the virtual machine,"* **in [0006] Lines 6-8**); and

sending the feedback (*see installation as feedback, e.g.* "*downloaded, and installed to the virtual machine,"* **in [0006] Lines 6-8**) from the server platform to the client platform through the network *(see communication from server to client through network*, **in Fig. 1**), but

Bantz does not teach a virtual machine identifier to identify a virtual machine in the client platform that is executing an application when the input operation happens as claimed.

However, **Takamura** teaches

the virtual machine (*see guest operating system ran in client,* **in Fig. 2 [122]**, *e.g. "The startup processing 320 is called when the client computer 101 is started and it activates the hypervisor and the OS,"* **in [0045] Lines 4-6**) is run on the client platform (*see "Client Computer,"* **in Fig. 1 [101]**).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the claimed invention to add the feature of running a Virtual Machine Monitor (VMM)/Hypervisor in a client computer to detect I/O requests that need to be handled by the Hypervisor of a host systems, as disclosed in **Takamura** into the teachings of **Bantz** since both of the references are directed to virtual machine operating system environments, Hence, would be considered to be analogous based on their related fields of endeavor, but

Bantz-Takamura does not teach

a virtual machine identifier to identify a virtual machine as claimed.

However, **Kaneda** teaches a virtual machine identifier (e.g. "*identification number,"* **in Col. 1, Lines 63**) to identify a virtual machine in the client (e.g. "*computer system,"* **in Col. 1, Lines 63-65**) platform that is executing the input operation (e.g. "*computer system for controlling virtual machines, each machine given a different identification number,"* **in Col. 1, Lines 63-65**).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the claimed invention to add the feature of virtual machine identification numbers as disclosed in **Kaneda,** into the teachings of **Bantz-Takamura** since all of the references are directed to virtual machine operating system environments, Hence, would be considered to be analogous based on their related fields of endeavor.

One would be motivated to combine **Takamura** with **Bantz** for reasoning set forth above in claim 1, and one would be motivated to combine **Kaneda** with **Bantz-Takamura** in order to specify which virtual machine running on the client is to receive feedback, as it should be obvious to one of ordinary skill in the art to recognize that some sort of identification is necessary when transferring data in a network to a particular endpoint that has a plurality of equivalent environments for that endpoint, as **Takamura** also discloses the use of multiple guest Operating Systems in as single computer platform (e.g. "*In an actual computer system, however, there are many cases that such an OS-based I/O device virtualization function is unusable. This is because the I/O device virtualization function is available only between identical operation systems, in many occasions, and further, a plurality of types of OS are mixed in one*

computer system in general," **from Takamura in [0007]**).

16. In regard to claim 15, **Bantz-Takamura-Kaneda** teaches the method of claim 14, wherein the feedback (*see installation as feedback, e.g.* "*downloaded, and installed to the virtual machine,"* **from Bantz in [0006] Lines 6-8**) further comprise a client platform identifier to identify the client platform that has sent the request (*see inherent client identifier to install the device driver on the virtual machine, e.g.* "*downloaded, and installed to the virtual machine,"* **from Bantz in [0006] Lines 6-8**).

17. In regard to claim 16, **Bantz-Takamura-Kaneda** teaches the method of claim 11, further comprising issuing an interrupt instruction (*e.g. "if an interrupt request is in that port, an I/O interrupt for the VM monitor of the real machine will be generated,"* **from Kaneda in Col. 4, Lines 20-22**) from a device module (*e.g. "the device driver to be located*," **from Bantz in [0006] Line 7**) of the plurality of device modules in the server platform (e.g. "*The device information is used to…find out if support for that particular device exists on the server,"* **from Bantz in [0027] Line 7 – [0028] Line 4**).

One would be motivated to combine **Bantz-Takamura** with **Kaneda** for reasoning set forth above in claim 5.

18. In regard to claim 17, **Bantz-Takamura-Kaneda** teaches the method of claim 11, wherein the interrupt instruction (e.g. "*an I/O interrupt,"* **from Kaneda in Col. 4, Lines 20-22**) further comprises a virtual machine identifier (e.g. "*identification number,"* **from**

Kaneda in Col. 1, Lines 63) to identify a virtual machine in the client platform to handle the interrupt (*e.g. "By this handling routine…it is determined which virtual machine has issued the I/O instruction which caused the I/O interrupt,"* **from Kaneda in Col. 6, Lines 40-43**).

One would be motivated to combine **Bantz-Takamura** with **Kaneda** for reasoning set forth above in claim 5.

19. Claims 23-25 are corresponding machine readable storage medium claims (*see "HDD,"* **in Fig. 1 [903] [913],** *e.g. "In the HDD 903, there are stored an application program 121, an operating system 122, a hypervisor 123, and a boot loader 124,"* **from Takamura in [0028]**) of method claims 15-17; therefore, they are rejected under the same rational.

20. In regard to claim 26, **Bantz** teaches a system, comprising

a client platform (*see client platform*, **in Fig. 3 [104])** comprising:

determining (*e.g. "recognized,"* **in [0006] Line 3**) that an input/output operation related to a hardware device (*e.g. "plugged in,"* **in [0006] Lines 2-3**) happens in a virtual machine (*e.g. "the device to be detected locally, the device driver to be located, downloaded, and installed to the virtual machine,"* **in [0006] Lines 6-8**) and construct a request for the input/output operation (e.g. "*find out if support for that particular device exists on the server,"* **in [0028] Lines 3-4**);

a client network interface (*see inherent communication interface to communicate with server,* **in Fig. 3 [101] [104]**) to send the request through a network (*see sending and receiving via network,* **in Fig. 1)**; and the server platform (*see server platform*, **in Fig. 1 [101])** comprising:

a server network interface (*see inherent communication interface to communicate with client,* **in Fig. 3 [101] [104]**) to receive the request through the network (*e.g. "sends that information to the virtual machine instance in server…The device information is used to…find out if support for that particular device exists on the server,"* **in [0027] Line 7 – [0028] Line 4**);

a plurality of device modules (*e.g. "the device driver to be located*," **in [0006] Line 7**);

a controller to identify a device module from the plurality of device modules (*e.g. "the device driver to be located*," **in [0006] Line 7**) to handle the request (*e.g. "find out if support for that particular device exists on the server…If not, the virtual machine instance in the server initiates the installation of a physical device driver in the server,"* **in [0028] Lines 3-6**), the identified device module is a virtual device corresponding to the input/output (*e.g. "The device information is used to allow the virtual machine to choose what type of device 103 has been inserted into the virtual device hub...the virtual machine instance in the server 101 initiates the installation of a physical device driver in the server 101 and a virtual device driver in the virtual machine instance executing on the server 101,"* **in [0027] Lines 5-8**),, but

Bantz does not teach

a virtual machine monitor to determine that an input/output operation related to the input/output device happens during execution of an application on a virtual machine of a plurality of virtual machines as claimed.

However, **Takamura** teaches

a virtual machine monitor (*see "Hypervisor,"* **in Fig. 2 [123]**) to determine that an input/output operation related to the input/output device happens (e.*g. "hypervisor of the client computer…for detecting an access to an I/O device of the server computer…and…transmitting a command to the I/O device of the server computer….A hypervisor of the server computer…which receives the command to the I/O device from the network, and issues the command to the I/O device,"* **in [0010] Lines 4-14**) during execution of an application (*e.g. "The application program 121 is a program including file reading 210 and file writing 360, and it carries out reading and writing from/to the I/O device 914, which is connected to the server computer 102,"* **in [0029] Lines 1-4**) on a virtual machine (*see guest operating system ran in client,* **in Fig. 2 [122]**, *e.g. "The startup processing 320 is called when the client computer 101 is started and it activates the hypervisor and the OS,"* **in [0045] Lines 4-6**), but

Bantz-Takamura does not explicitly teach

a plurality of virtual machines as claimed.

However, **Kaneda** teaches

a plurality of virtual machines (*e.g. "virtual machines each given a different identification number,"* **from Abstract**).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the claimed invention to combine **Bantz-Takamura-Kaneda** for reasoning set forth above in claim 11

21. In regard to claim 27, **Bantz-Takamura-Kaneda** teaches the system of claim 26, wherein the request (e.g. "*find out if support for that particular device exists on the server,"* **from Bantz in [0028] Lines 3-4**) further comprises a device module identifier to identifier the device module in the server platform (*see inherent identification of server platform in connection of client to the server,* **from Bantz in Fig. 1 [101] [104]**).

22. In regard to the system of claim 28, **Bantz-Takamura-Kaneda** teaches wherein the identified device module in the server platform is further to obtain a result (*e.g. "recognized,"* **from Bantz in [0006] Line 3**) for the input/output operation (*e.g. "the device to be detected locally,"* **from Bantz in [0006] Lines 6-8**), and construct a feedback with the result (*see installation as feedback, e.g.* "*downloaded, and installed to the virtual machine,"* **from Bantz in [0006] Lines 6-8**) and a virtual machine identifier (e.g. "*identification number,"* **from Kaneda in Col. 1, Line 63**) to identify the virtual machine in the client platform (e.g. "*computer system,"* **from Kaneda in Col. 1, Lines 63-65**) under control from the controller (e.g. "*computer system for controlling virtual machines, each machine given a different identification number,"* **from Kaneda in Col. 1, Lines 63-65**),

and the server network interface (*see inherent communication interface to communicate with client,* **from Bantz in Fig. 1 [101] [104]**) is further to send the feedback to the client platform through the network (*see server sending the device driver through the network to the virtual machine on client,* **in Fig. 1**, *e.g.* "*downloaded, and installed to the virtual machine,"* **from Bantz in [0006] Lines 6-8**).

One would be motivated to combine **Bantz-Takamura** with **Kaneda** for reasoning set forth above in claim 11.

23. In regard to claim 29, **Bantz-Takamura-Kaneda** teaches the system of claim 26, wherein

the client network interface (*see inherent communication interface to communicate with server,* **from Bantz in Fig. 1 [101] [104]**) is further to receive a feedback for the input/output operation from the server platform through the network (*see server sending the device driver through the network to the virtual machine on client,* **in Fig. 1**, *e.g.* "*downloaded, and installed to the virtual machine,"* **from Bantz in [0006] Lines 6-8**); and

the virtual machine monitor (*e.g. "the VM monitor,"* **from Kaneda in Abstract**) is further to identify the virtual machine in the client platform that is executing the input/output operation (*e.g. "executes a program of the VM monitor…to transfer the control right of the CPU to one of the programs of the virtual machine regions…allocated for each virtual machine, so that one virtual machine may be operated,"* **from Kaneda in Col. 3, Lines 50-54**) based upon the feedback and send

the feedback to the identified virtual machine (*see installation as feedback, e.g.* "*downloaded, and installed to the virtual machine,"* **from Bantz in [0006] Lines 6-8**).

One would be motivated to combine **Bantz-Takamura** with **Kaneda** for reasoning set forth above in claim 11.

24. In regard to claim 30, **Bantz-Takamura-Kaneda** teaches the system of claim 26, wherein

a device module (*e.g. "the device driver to be located,"* **from Bantz in [0006] Line 7**) in the server platform (e.g. "*The device information is used to…find out if support for that particular device exists on the server,"* **from Bantz in [0027] Line 7 – [0028] Line 4**) is to issue an interrupt instruction under control from the controller (*e.g. "if an interrupt request is in that port, an I/O interrupt for the VM monitor of the real machine will be generated,"* **from Kaneda in Col. 4, Lines 20-22**), the interrupt instruction including a virtual machine identifier to identify another virtual machine in the client platform to handle the interrupt instruction (*e.g. "By this handling routine…it is determined which virtual machine has issued the I/O instruction which caused the I/O interrupt,"* **from Kaneda in Col. 6, Lines 40-43**); and

the server network interface (*see inherent communication interface to communicate with client,* **from Bantz in Fig. 1 [101] [104]**) is further to send the interrupt instruction (e.g. "*I/O interrupt"* **from Kaneda in Col. 4, Lines 20-21**) to the client platform through the network (*see connection from server to client*, **from Bantz in Fig. 1 [101] [104]**).

One would be motivated to combine **Bantz-Takamura** with **Kaneda** for reasoning set forth above in claim 5.

25. In regard to claim 31, **Bantz-Kaneda** teaches the system of claim 30, wherein

the client network interface *see inherent communication interface to communicate with server,* **from Bantz in Fig. 1 [101] [104]**) is further to receive the interrupt instruction (*see connection from server to client*, **from Bantz in Fig. 1 [101] [104]**); and

the virtual machine monitor (*e.g. "the VM monitor,"* **from Kaneda in Abstract**) is further to identify the another virtual machine (*e.g. "By this handling routine…it is determined which virtual machine has issued the I/O instruction which caused the I/O interrupt,"* **from Kaneda in Col. 6, Lines 40-43**) from the plurality of virtual machines (*e.g. "virtual machines each given a different identification number,"* **from Kaneda in Abstract**) based upon the interrupt instruction and inject (*e.g. "By this handling routine," in Col. 6, Line 40*) the interrupt into the identified another virtual machine (*e.g. "By this handling routine…it is determined which virtual machine has issued the I/O instruction which caused the I/O interrupt,"* **from Kaneda in Col. 6, Lines 40-43**).

One would be motivated to combine **Bantz-Takamura** with **Kaneda** for reasoning set forth above in claim 5.

26. Claims 35-37 recite claims that contain substantially the same limitations of claims 14-16; therefore, they are rejected under the same rational.

27. In regard to claim 38, **Bantz-Takamura** teaches the method of claim 32, but **Bantz-Takamura** does not teach

wherein the interrupt instruction further comprising a virtual machine identifier to identify another virtual machine in the client machine to handle the interrupt instruction as claimed.

However, **Kaneda** teaches:

interrupt instruction (e.g. "*I/O interrupt*" **in Col. 4, Lines 20-21**) comprising a virtual machine identifier (*e.g. "identification number,*" **in Col. 6, Line 1**) to identify another virtual machine to perform the interrupt instruction (*e.g. "By this handling routine…it is determined which virtual machine has issued the I/O instruction which caused the I/O interrupt,"* **in Col. 6, Lines 40-43**).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the claimed invention to combine **Bantz-Takamura** with **Kaneda** for reasoning set forth above in claim 5.

28. In regard to claim 39, **Bantz-Takamura-Kaneda** teaches the method of claim 38, further comprising:

receiving an interrupt instruction (*e.g. "if an interrupt request is in that port, an I/O interrupt for the VM monitor of the real machine will be generated,"* **from Kaneda in Col. 4, Lines 20-22**) through the network by the client platform (*e.g. "recognized,"* **from Bantz in [0006] Line 3**)

identifying the another virtual machine in the client platform based upon the interrupt instruction (*e.g. "By this handling routine…it is determined which virtual machine has issued the I/O instruction which caused the I/O interrupt,"* **from Kaneda in Col. 6, Lines 40-43**); and

injecting the interrupt instruction (*e.g. "By this handling routine," in Col. 6, Line 40*) into the identified another virtual machine (*e.g. "By this handling routine…it is determined which virtual machine has issued the I/O instruction which caused the I/O interrupt,"* **from Kaneda in Col. 6, Lines 40-43**).

One would be motivated to combine **Bantz-Takamura** with **Kaneda** for reasoning set forth above in claim 5.

29. Claims 13, 21, and 33-34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bantz-Takamura in view of US 2005/0198303 A1 to Knauerhase et al. (hereinafter referred to as Knauer).

30. In regard to claim 13, **Bantz-Takamura** teaches the method of claim 11, but **Bantz-Takamura** does not teach

determining whether the identified device module is in another server platform; and

sending the request from the server platform to the another server platform via the network, in response to determining that the identified device module is in the another server platform as claimed.

However, **Knauer** teaches determining (*e.g. "the server determines if a virtual machine already exists that offers the service,"* **in Abstract**) whether the identified device module (*e.g. "service from the virtual machine,"* **from Abstract**) is in another server platform (*see plurality of servers hosting virtual machines,* **in Fig. 1 [125]**, *e.g. "server is coupled to carious other servers in server farm,"* **in [0020] Lines 1-2**); and

sending the request from the server platform to the another server platform via the network (*e.g. "see servers coupled together through network,"* **in Fig. 1**), in response to determining that the identified device module is in the another server platform (e.*g. "the server determines if the requested service may be offered…the server switches, based on whether the requested service may be offered,"* **in [0047] Lines 11-14**).

Therefore it would have been obvious to one of ordinary skill in the art at the time of the current invention to add the feature of determining an additional server to obtain a service for handling a request as disclosed in **Knauer**, into the teachings of **Bantz-Takamura**, since all of the references are directed to providing services to virtual machine operating system environments, hence, would be considered to be analogous based on their related fields of endeavor.

One would have been motivated to do so to add the additional benefit of having a backup server in case a primary server did not have the required software or was unable to fulfill a request in a desired way, as **Knauer** discloses the need for providing services to user's in different operating system environments (e.*g. "to offer other services requiring a different, incompatible hosting environment (e.g. different operating*

system or supporting environment software versions), the service provider has to configure another server with the other services…The invention addresses these problems and others in the art," **from Knauer in [0005] - [0006]**)

31. Claim 21 is a corresponding machine readable storage medium claim (*see "HDD,"* **in Fig. 1 [903] [913],** *e.g. "In the HDD 903, there are stored an application program 121, an operating system 122, a hypervisor 123, and a boot loader 124,"* **from Takamura in [0028]**) of method claim 13; therefore, it is rejected under the same rational.

32. Claims 33-34 recite claims that contain substantially the same limitations of claim 13; therefore, they are rejected under the same rational.

Response to Arguments

33. In the Arguments/Remarks Applicant's argued in substance that:

(A) Computer readable medium is regarded as a statutory subject matter under a PTO guideline for 101. **(Page 15)**

As to Argument A, Examiner respectfully disagrees with applicants because given the broadest reasonable interpretation in light of the specification and taking into

account the meaning of the words in their ordinary usage as they would be understood by one of ordinary skill in the art (MPEP §2111), the claim as a whole covers both transitory and non-transitory media, and a transitory medium does not fall into any of the 4 categories of invention (process, machine, manufacture, or composition of matter).

(B) Bantz-Takamura does not teach a request comprising a device module identifier to identify a device module from a plurality of device modules in the server platform to handle the input/output operation, wherein the device module is a virtual device corresponding to the input/output device, because in **Bantz** the server is installed with the virtual device driver which is different from a virtual device corresponding to the I/O device, namely the device module. **(Pages 16-17)**

As to Argument B, Examiner respectfully disagrees with applicants as clearly **Bantz** teaches a request comprising a device identifier, in order to identify a device driver from a plurality of device drivers (e.*g. "gathers the information about the device such as the device model number and type, and sends that information to the virtual machine instance in server,"* **from Bantz in [0027] Lines 5-8**) and **Takamura** teaches a client running it's own virtual machine detecting I/O events (e.*g. "hypervisor of the client computer…for detecting an access to an I/O device,"* **from Takamura in [0010] Lines 4-14**), and it is unclear exactly what Applicant is arguing by:

(1) In **Bantz** "the server is installed with the virtual device driver which is different from a virtual device corresponding to the I/O device, namely the device module."

(2) "**Bantz** still teaches that the device module established by the device virtualization layer should be installed in the user, rather than the server."

(3) "**Takamura** does not teach anything about device module, except for Line 7 of paragraph 0029 teaching that when an application running on a client computer carries out a read or write command, the memory protection interrupt processing 300 of the client computer detects the read or write command to the logical I/O device."

(C) "**Bantz** does not teach determining that an input/output operation related to an input/output device happens during execution of an application on a virtual machine of the client platform. Moreover, even though as the Office Action states that the above "determining" means that a virtual machine was performing the execution, the above "determining" clearly states that the execution is performed on the virtual machine of the client platform, which nothing from **Bantz** teaches." **(Pages 17-18)**

As to Argument C, Examiner would like to note that one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.,* 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986), and as previously stated in the prior Office Action, Examiner is not relying on the **Bantz** reference to teach that the virtual machine is run on the client device, but the combination of **Bantz-Takamura** teaches a virtual machine running on a client machine that detects I/O operations (e.*g. "hypervisor of the client computer…for detecting an*

access to an I/O device of the server computer…and…transmitting a command to the I/O device of the server computer….A hypervisor of the server computer…which receives the command to the I/O device from the network, and issues the command to the I/O device," **from Takamura in [0010] Lines 4-14**).

(D) **Bantz** and **Takamura** don not present a *Prima Facie* case of obviousness to combine because the references conflict one another because in **Bantz** the devices are local to the user and in **Takamura** the devices are remote to the sure, and **Bantz** is directed toward a thin client so **Bantz** would have no reason to use a hypervisor as disclosed in **Takamura**. **(Pages 18-20)**

As to argument D, Examiner respectfully disagrees with Applicants, noting that the examiner recognizes that obviousness may be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988), *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992), and *KSR International Co. v. Teleflex, Inc.*, 550 U.S. 398, 82 USPQ2d 1385 (2007).

In this case, **Bantz** is directed toward recognizing local peripheral devices connected to a virtual machine of a client and supplying support for the local peripheral devices from a remote server, and **Bantz** also discloses the invention is applicable to

physical devices as well as virtual devices such as a fat client (*e.g. "The invention is applicable to physical machines as well as virtual machines as shown in FIG. 3. When the invention is applied to a physical machine, a device virtualization layer 301 is interposed in the operating system between the device 103 and the server 101,"* **from Bantz in [0033] and** *e.g. "The layer, in turn, implements communication with real devices through local device drivers or through device drivers that communicate with the device virtualization layer of some remote platform,"* **from Bantz in [0012] Lines 6-9**), wherein in the physical implementation a Virtualization Layer (similar to a Virtual Machine Monitor or Hypervisor) is inserted between the peripheral devices and the remote server, but the **Bantz** reference does not go into full detail about such an implementation; However, the **Takamura** references uses a similar implementation, wherein a Hypervisor is run on a client to detect logical device operations, and one of ordinary skill in the art would have been motivated by the combination of **Bantz** and **Takamura** to detect any logical operations, either local or remote, on the Virtualization Layer/Hypervisor of a physical client device to provide local user's the ability to utilize such a system without the need to be constantly connected to a remote server and prevent all of the common disadvantages of an "always connected" device, such as extra costs accumulated for connectivity and power.

(E) Applicant fails to see anything from **Bantz** that teaches a client system can be fat and unsupported drivers can be stored on the fat client from the Server. **(Page 6)**

As to argument E, Examiner respectfully disagrees with Applicants, as one of ordinary skill in the art would recognize that **Bantz** could operate in a thin client or fat client system, thereby necessitating installation of unsupported drivers on a fat client from a remote server, because it is well known in the art that Virtual Machines are commonly run locally as well as remotely, depending on the processing and storage needs of a particular system, in which is supported by **Bantz**, since **Bantz** is directed toward a thin approach and a fat approach (*e.g. "The invention is applicable to physical machines as well as virtual machines as shown in FIG. 3. When the invention is applied to a physical machine, a device virtualization layer 301 is interposed in the operating system between the device 103 and the server 101,"* **from Bantz in [0033]**).

Conclusion

34. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

US 5,996,026 to Onodera et al.

US 6,418,464 B1 to Minow

US 2003/0090704 A1 to Hansen

US 2005/0076324 A1 to Lowell et al.

US 2003/0208642 A1 to Desai et al.

US 2005/0076155 A1 to Lowell

35. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

36. Any inquiry concerning this communication or earlier communications from the examiner should be directed to JONATHAN WILLIS whose telephone number is (571)270-7467. The examiner can normally be reached on 8:00 A.M. - 6:00 P.M..

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Andrew Caldwell can be reached on (571)272-3868. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/JW/
Jonathon Willis
Examiner, Art Unit 2445
1/10/2011

/Andrew Caldwell/
Supervisory Patent Examiner, Art Unit 2445